often referred to as garbage collection, to designate memory as "free" when it is no longer needed for its current allocation.--

Please replace the paragraph on page 10, lines 4-14 with the following:

A2

--Returning to Figure 1A, the input to the method of the invention is the destination PC for the method and the storage area destination to which the resulting information on the stack shape will be written (block 100). When the mapping occurs at runtime, the definition of the storage destination will point to a location on the stack; when the mapping occurs at compile time, the pointer will be into an array for storage with the compiled method on the heap. The different uses of the invention for stack mapping at runtime and at compilation are discussed in greater detail below.--

Please replace the paragraph on page 13, line 28 through page 14, line 4 with the following:

A3

--Temporary fetch and store instructions are normally one or two bytes long. One byte is for the bytecode and one byte is for the parameter unless it is inferred by the bytecode. However, Java includes an escape sequence which sets the parameters for the following bytecode as larger than normal (wide bytecode). This affects the stack mapper only in how much the walk count is incremented for the next byte. It does not affect control.--

Please replace the paragraph on page 17, line 23 through page 18, line 9 with the following:

A4

--The compressed encoded stack map is stored statically in the compiled method or on the